Clipping Chaos to Cycles

Sudeshna Sinha
The Institute of Mathematical Sciences,
Chennai, India

Homepage: http://www.imsc.res.in/∼sudeshna

A strategy for extracting regular temporal patterns in a controlled manner from chaotic dynamics

Enables us to harness the richness of chaos in a direct and efficient way

Principle : Restricts available phase space

Dynamic Range Limiter

★ Prunes chaotic temporal sequences to stable desired patterns

★ Chaos advantageous as it possesses a rich range of temporal patterns which can be clipped to different behaviours

Consider a general dynamical system, and choose a state variable to be monitored

Threshold Mechanism is triggered whenever the value of the variable exceeds a critical threshold x^*

The variable is then re-set to x^*

If $x > x^*$ then $x = x^*$

The dynamics continues till the next occurrence of the variable exceeding the threshold

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

- $x^* < 0.5$: Fixed point

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

- $x^* < 0.5$: Fixed point
- $0.5 < x^* < 0.809$: Period 2

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

- $x^* < 0.5$: Fixed point
- $0.5 < x^* < 0.809$: Period 2
- $0.809 < x^* < 0.85$: Period 4

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

- $x^* < 0.5$: Fixed point
- $0.5 < x^* < 0.809$: Period 2
- $0.809 < x^* < 0.85$: Period 4
- $x^* = 0.86$: Period 6

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

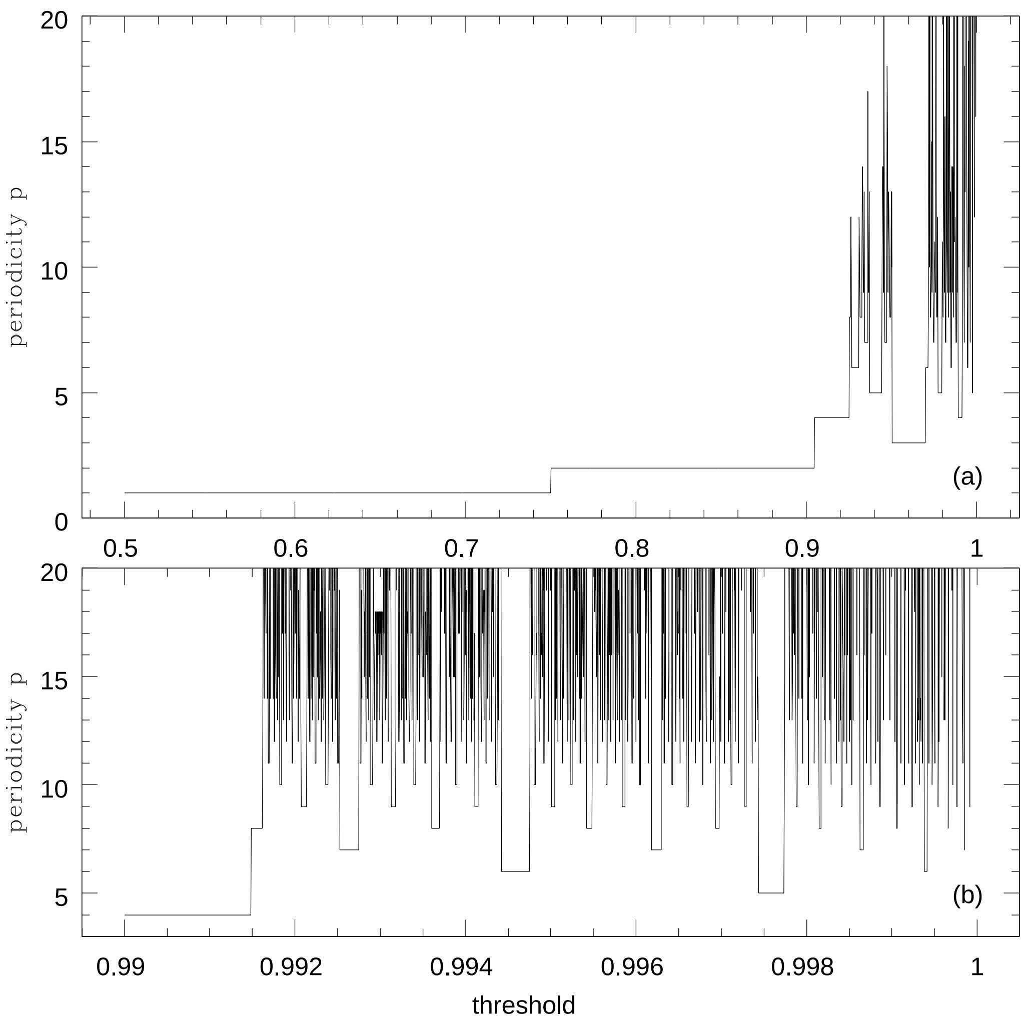
- $x^* < 0.5$: Fixed point
- $0.5 < x^* < 0.809$: Period 2
- $0.809 < x^* < 0.85$: Period 4
- $x^* = 0.86$: Period 6
- $x^* = 0.88$: Period 7

Different regular dynamical patterns obtained for different thresholds

For example for the chaotic logistic map $f(x) = 4x(1-x)$

- $x^* < 0.5$: Fixed point
- $0.5 < x^* < 0.809$: Period 2
- $0.809 < x^* < 0.85$: Period 4
- $x^* = 0.86$: Period 6
- $x^* = 0.88$: Period 7
- $x^* = 0.9$: Period 9

The Controlled Period – Threshold Correspondence



- Exact relations for the position and width of the periodic windows in threshold parameter space :

  Provides a look-up table to directly extract widely varying temporal patterns

- Yields a wide range of response patterns from the same module

  Thus useful for designing components that can switch flexibly between different behaviours

- Requires no run-time computations
- Transience is extremely short; Very robust
- Controller simple

Analysis

- Directly calculate the period corresponding to a certain threshold

Analysis

- Directly calculate the period corresponding to a certain threshold
- Answer the reverse (important) question as well: what threshold do we need to set in order to obtain a certain period

Be-heading the Chaotic Map

- Study the forward iterates of the map with initial value at threshold: $f(x^*), f^2(x^*), \ldots$
- Ascertain which iterate exceeds the threshold
- If the $k^{\rm th}$ iterate exceeds the threshold then we obtain period k
- Formulate the different solutions using the inverse map: L and R

Starting point of the analysis : the chaotic system, being ergodic, is guaranteed to exceed threshold at some point in time, at which point its state is re-set to x^*

One then studies the forward iterations of the map, starting from this state $x = x^*$, i.e.

$$f_0(x^*), f_1(x^*) \ldots$$

where $f_k(x^*)$ is the k^{th} iterate of the map

Specifically for the logistic map $f(x) = 4x(1-x)$:

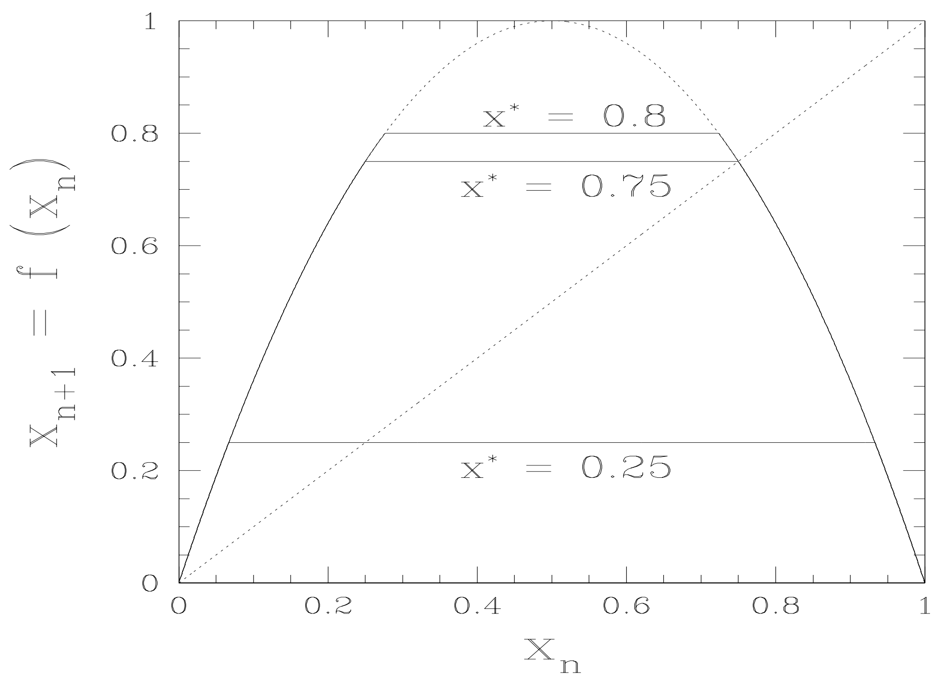
- $k = 0$; $f_0(x^*) = x^*$
- $k = 1$; $f_1(x^*) = 4x^*(1-x^*)$
- $k = 2$; $f_2(x^*) = 4(4x^*(1-x^*))(1-4x^*(1-x^*))$

In general

$$f_k(x^*) = f \circ f_{n-1}(x^*) = f \circ f \circ \ldots f \circ (x^*)$$

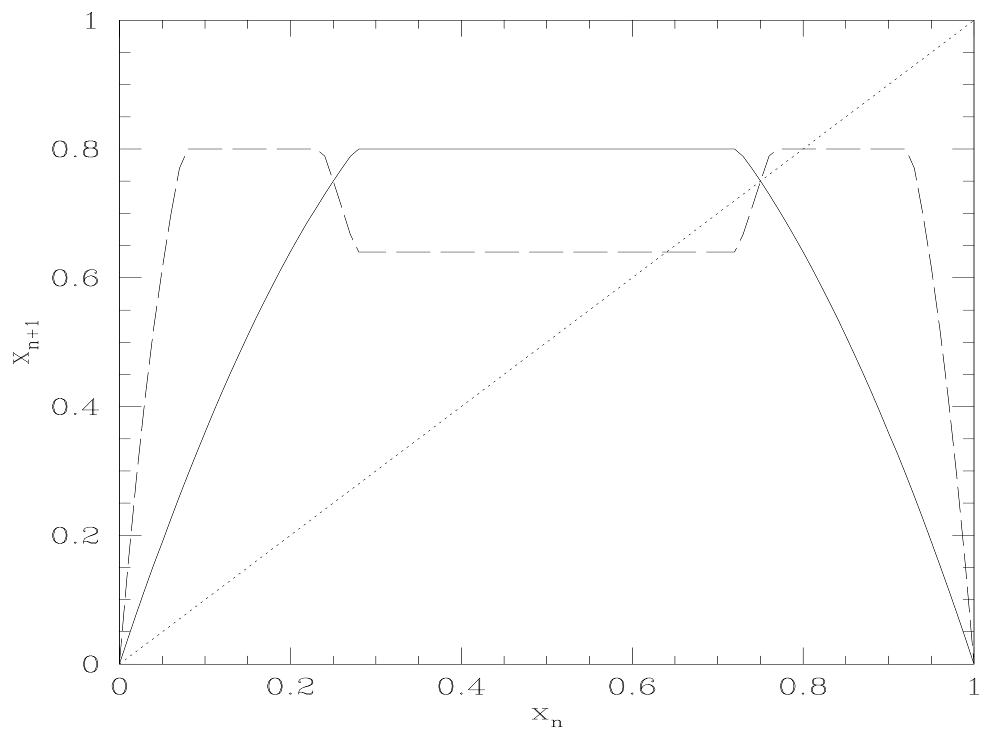
where threshold value $0 < x^* < 1$

First iterate x_{n+1} (—) of the effective thresholded map for different thresholds x^*



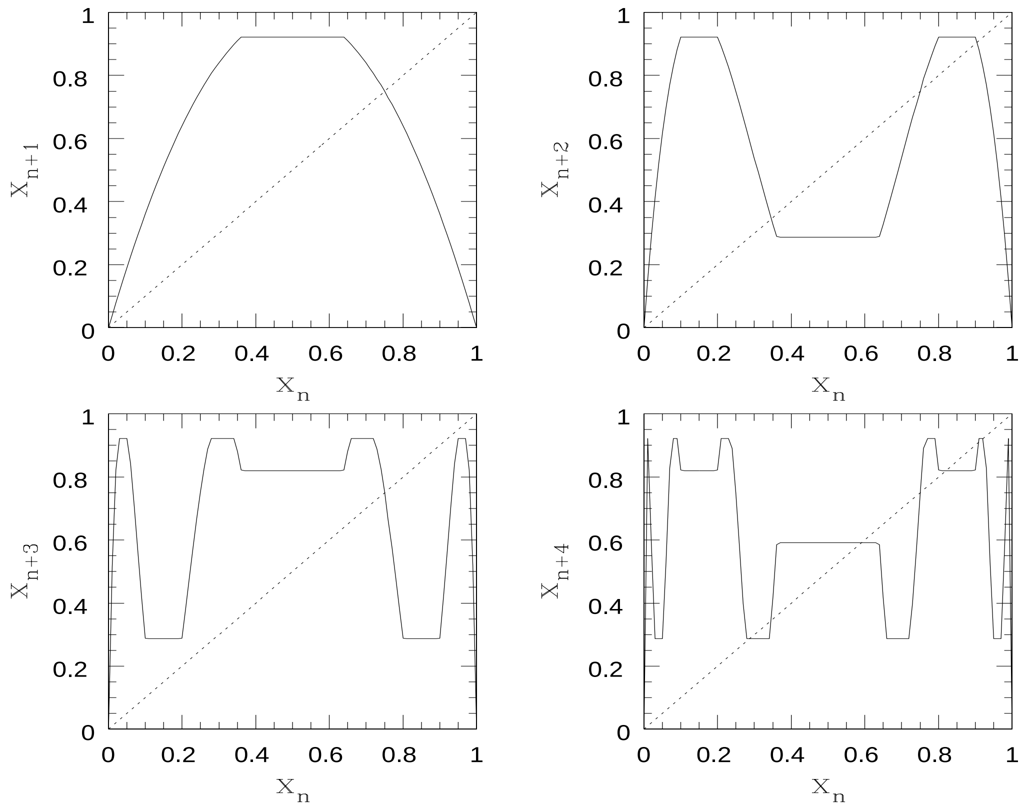
The intersection of the flat portion of the map x_{n+1} with the 45^0 line yields a superstable fixed point of period 1

Iterates x_{n+1} (—) and x_{n+2} (- - -) of chaotic map under thresholding : $x^* = 0.8$



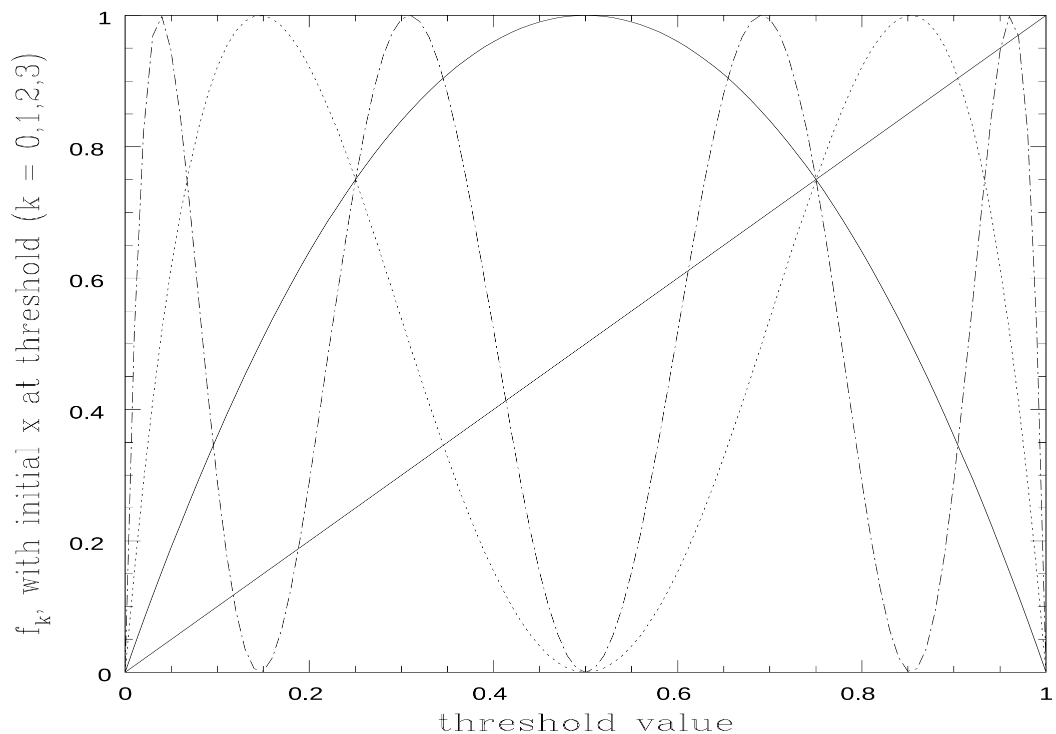
The intersection of the flat portion of the map x_{n+2} with the 45^0 line yields a superstable fixed point of period 2

Threshold value : 0.922



Intersection of the flat portion of the map x_{n+4} with the 45^0 line yields a superstable fixed point of period 4

Forward iterates of the chaotic logistic map starting from the threshol d value x^*



First 3 iterates of the map f_k : $k = 1$ (—), $k = 2$ (....), $k = 3$ (-.-.-) and $f_0(x^*) = x^*$ (—) (45^0 line)

When the $f_k(x^*)$ curve lies above the $f_0(x^*) = x^*$ line we have a k cycle : as this implies that the k^{th} iterate exceeds the critical value x^* and is re-set to x^*

$x^* = f_0(x^*)$ is the first point in the cycle

k - Cycle : $x^*,\ f_1(x^*),\ f_2(x^*),\ \ldots\ f_{k-1}(x^*)$

For instance, in the range $0 \leq x^* \leq \frac{3}{4}$ the $f_1(x^*)$ curve lie above the f_0 curve (i.e. $f_1(x^*) > x^*$)

So the chaotic element is adapted back to x^* at every iterate, yielding a period 1 fixed point

- In the range $\frac{3}{4} < x^* < 0.9$ the $f_1(x^*)$ curve dips below the 45^0 line, but the $f_2(x^*)$ curve lies above the 45^0 line
- This imples that the second iterate of the map (starting from $x = x^*$) exceeds threshold and is adapted back to x^*, thus giving rise to a period 2 cycle
- Thus the cycle at each value of threshold is the smallest k such that the k^{th} iterate of the map (starting from $x_0 = x^*$) is greater than x^*, i.e.

$$f_k(x^*) > x^*$$

- The chaotic element can then yield a wide variety of dynamical behaviour determined by the threshold

- In this manner the threshold mechanism leads to regular cyclic evolution, whose period depends on the threshold
- Thus in threshold parameter space we can find windows of various periods
- These are intervals where the following equation is satisfied:

  Period $P(x^*) = k$ iff $f_k(x^*) \geq x^*$ and $f_l(x^*) < x^*$ for all $l < k$.
- $P(x^*)$ is a piecewise continuous function of x^*

- For every cycle of periodicity k there will be several windows
- Upper bound of 2^{k-1} windows for period k
- The “middle” of the period k windows lies approximately where the curve $f_k(x^*)$ touches 1 (since if it touches 1 it has to have exceeded x^*, as $x^* < 1$)
- Then the solutions of the equation $f_k(x^*) = 1$ gives the x^* values corresponding to a period k

- The solutions can be formulated as:

$$f^{-1} \circ f^{-1} \circ f^{-1} \circ f^{-1}(1)$$

where f^{-1} is the (double valued) inverse map :

$$f^{-1}(y) = \frac{1}{2} \pm \frac{\sqrt{1-y}}{2}$$

- This has two values : on the right of the centre (denoted as R) and on the left of the centre (denoted as L)
- For $f^{-1}(1)$: $L(1) = R(1) = \frac{1}{2}$
For all other values : $L < R$

- Number of distinct values arising from the expression $f^{-1} \circ f^{-1} \ldots f^{-1}(1)$ is 2^{k-1}
- These arise from the 2^{k-1} different possible combinations of R and L
- The evaluation of this algebraic expression for various values of k is simple and direct

- The existence of a window of period k ($k > 1$) is dependent on the pervious iterates as well, i.e. a solution for period k may be masked by the fact that some iterate l, $l < k$, may have $f_l(x^*) > x^*$
- For instance for $k > 1$ all combinations starting with symbol L are masked by period 1 (as the period 1 window extends from 0 to $\frac{3}{4}$ and $L(x) \leq \frac{1}{2}$)
- So half of the combinations of $f^{-1} \circ f^{-1} \ldots f^{-1}(1)$ are swallowed by period 1
- One has to examine the remaining 2^{k-2} combinations to check which ones survive masking by lower order windows.

- Note that one family of windows is guaranteed to exist:

$$RL^{k-1}(1)$$

as all iterates leading up to 1 here, namely all the subsequences $L(1)$, $L^2(1)$, ... $L^{k-1}(1)$, have value less than $\frac{1}{2}$ (as they are all composed of L)

- Since all relevant thresholds for $k > 1$ are greater than $\frac{3}{4}$ it implies that all the iterates leading up to $f_k(x^*)$ have value less than x^* and so this sequence will always yield period k (not any other lower period)
- So all possible periods k have atleast one stable window in threshold space

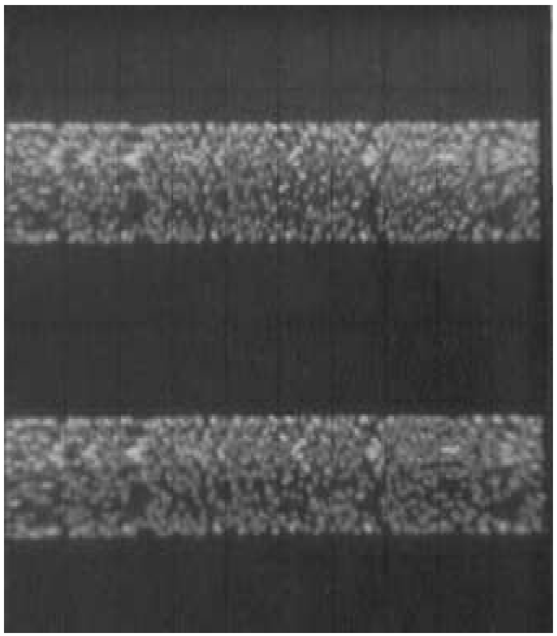
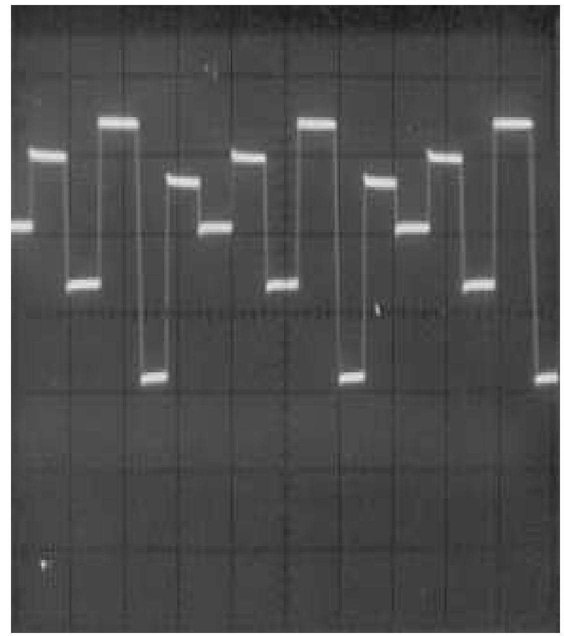
For chaotic maps it can then be analytically shown :

- Threshold control yields periods of all orders
- The system is trapped in a super-stable cycle the instant it exceeds threshold

- Thresholding clips chaos to desired time sequences
- Periodicity enforced on the sequences : thresholding acts as a re-setting of initial conditions

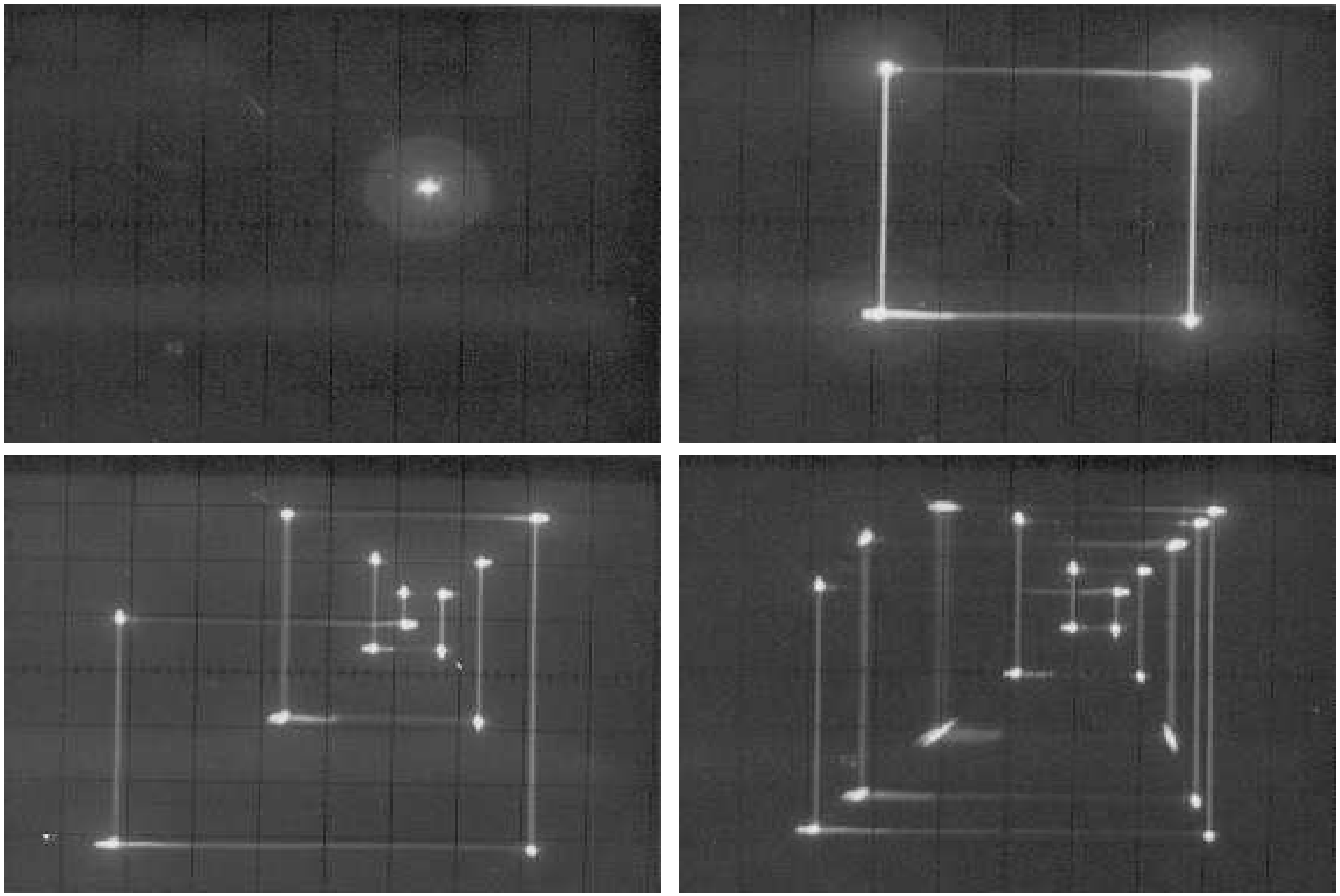
Ref: Sudeshna Sinha,
Physical Review E, 1993; Physics Letters A, 1994;
Also reviewed in Int. J. of Modern Physics, 1995

Experimental verification of clipping chaos to periods of wide ranging orders

 Chaotic Trace  6 - Cycle

Circuit Realization of the Logistic Map

Murali, Sinha and Ditto, Physical Review E, 2003



Complete agreement with theoretical analysis

Does thresholding work beyond iterative 1d maps?

Can continuous time higher dimensional (possibly hyper-chaotic) systems be clipped?

No exact results : must rely on numerics and experimentation

Nonlinear third order ordinary differential equations

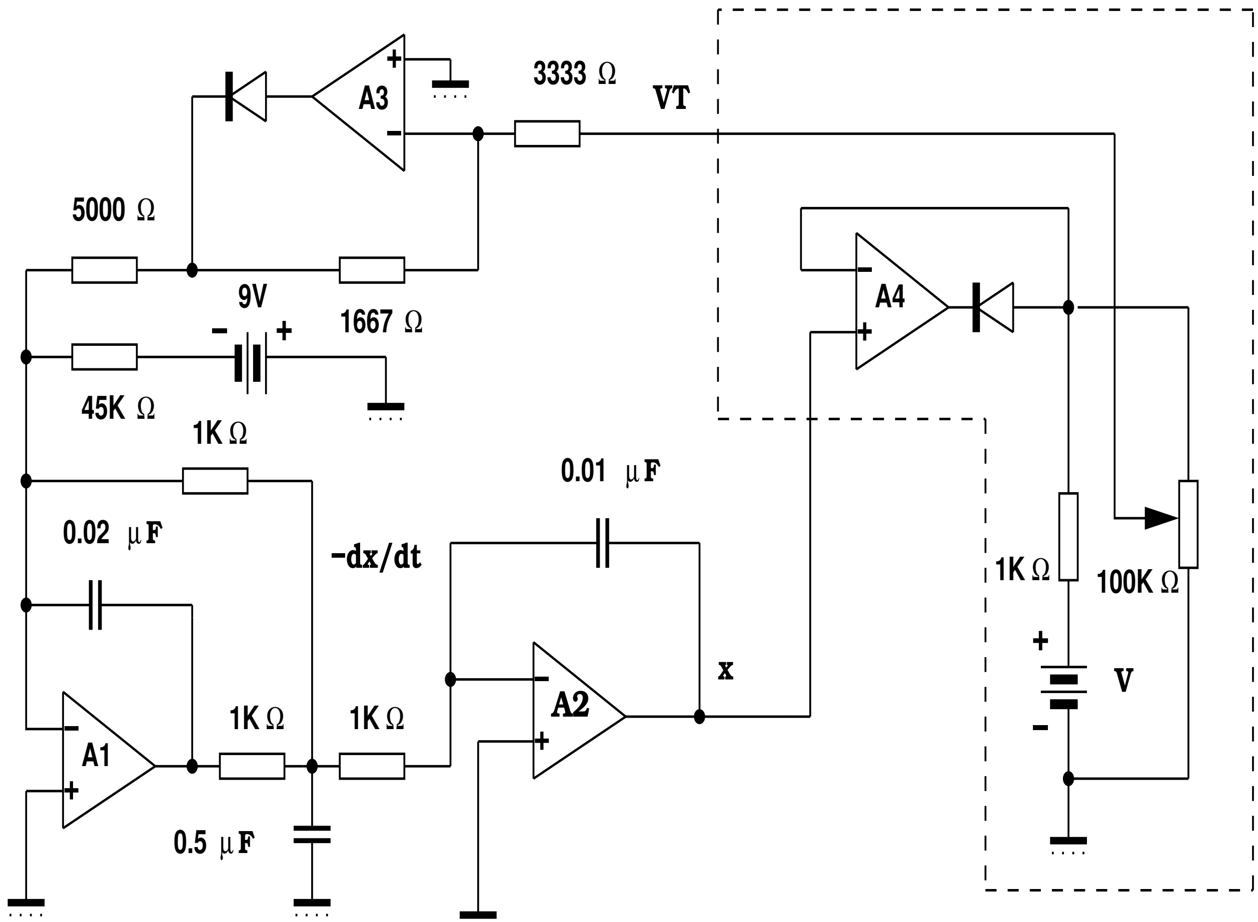
$$\frac{d^3x}{dt^3} + A\ \frac{d^2x}{dt^2} + \frac{dx}{dt} = G(x)$$

where $G(x)$ is a piecewise linear function:

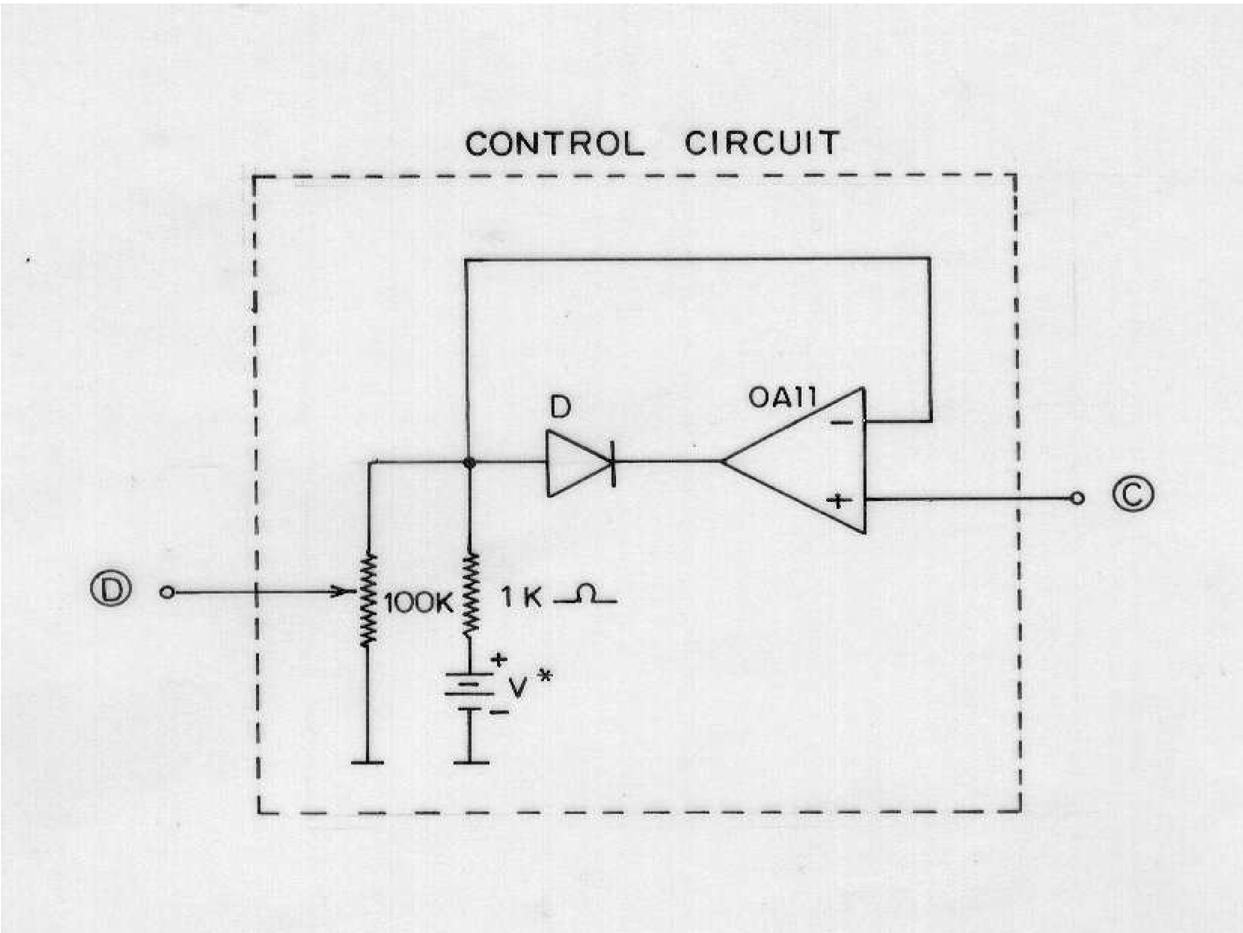
$G(x) = B|x| - C$

with $B = 1.0, C = 2.0$ and $A = 0.6$

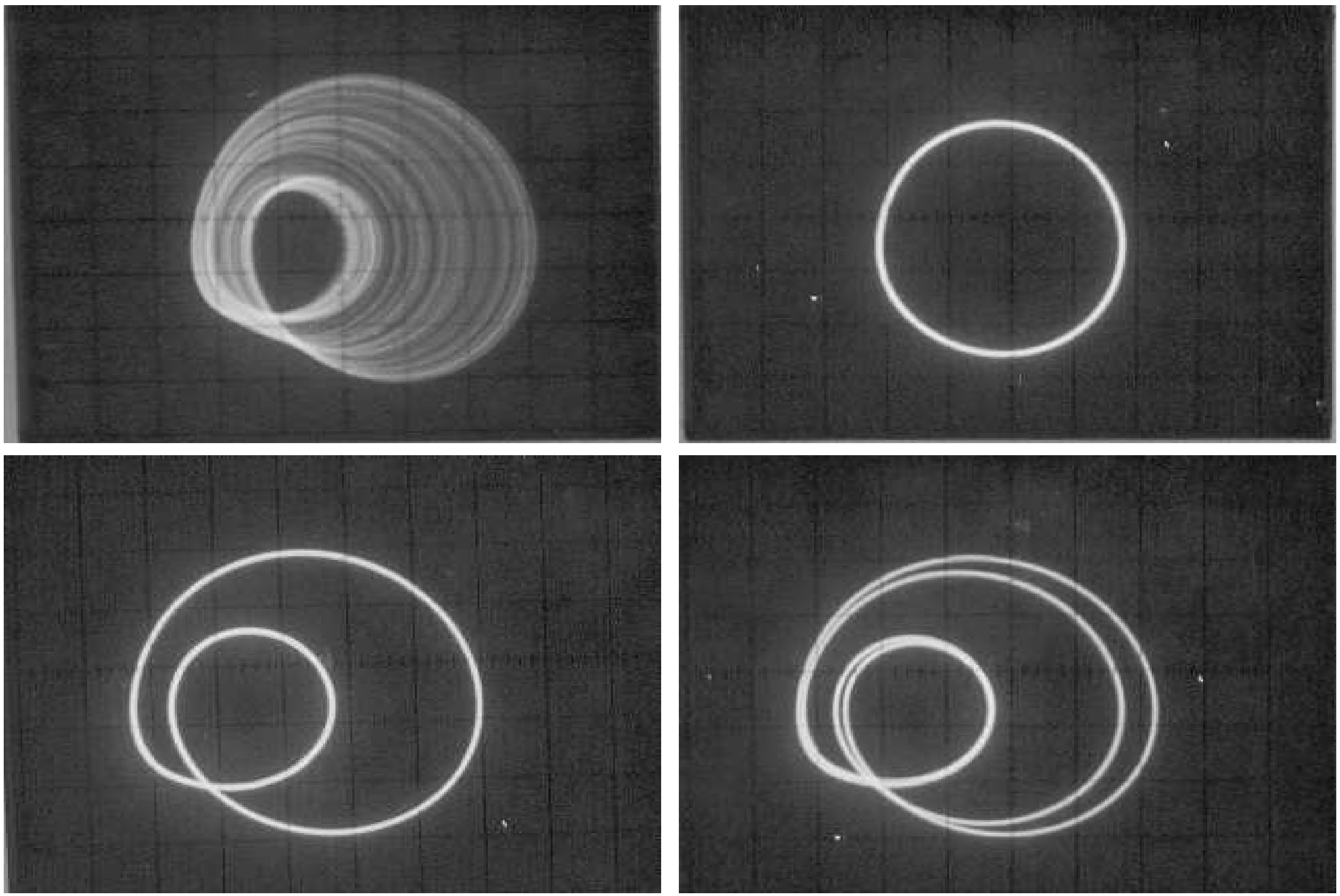
The circuit realisation of the above uses resistors, capacitors, diodes and operational amplifiers



Precision Clipping Circuit for Thresholding



Circuit realization of coupled third order nonlinear differential equations



Double scroll chaotic Chua's attractor given by the following set of (rescaled) 3 coupled ODEs

(1) $$\dot{x} = \alpha(y - x - g(x))$$

(2) $$\dot{y} = x - y + z$$

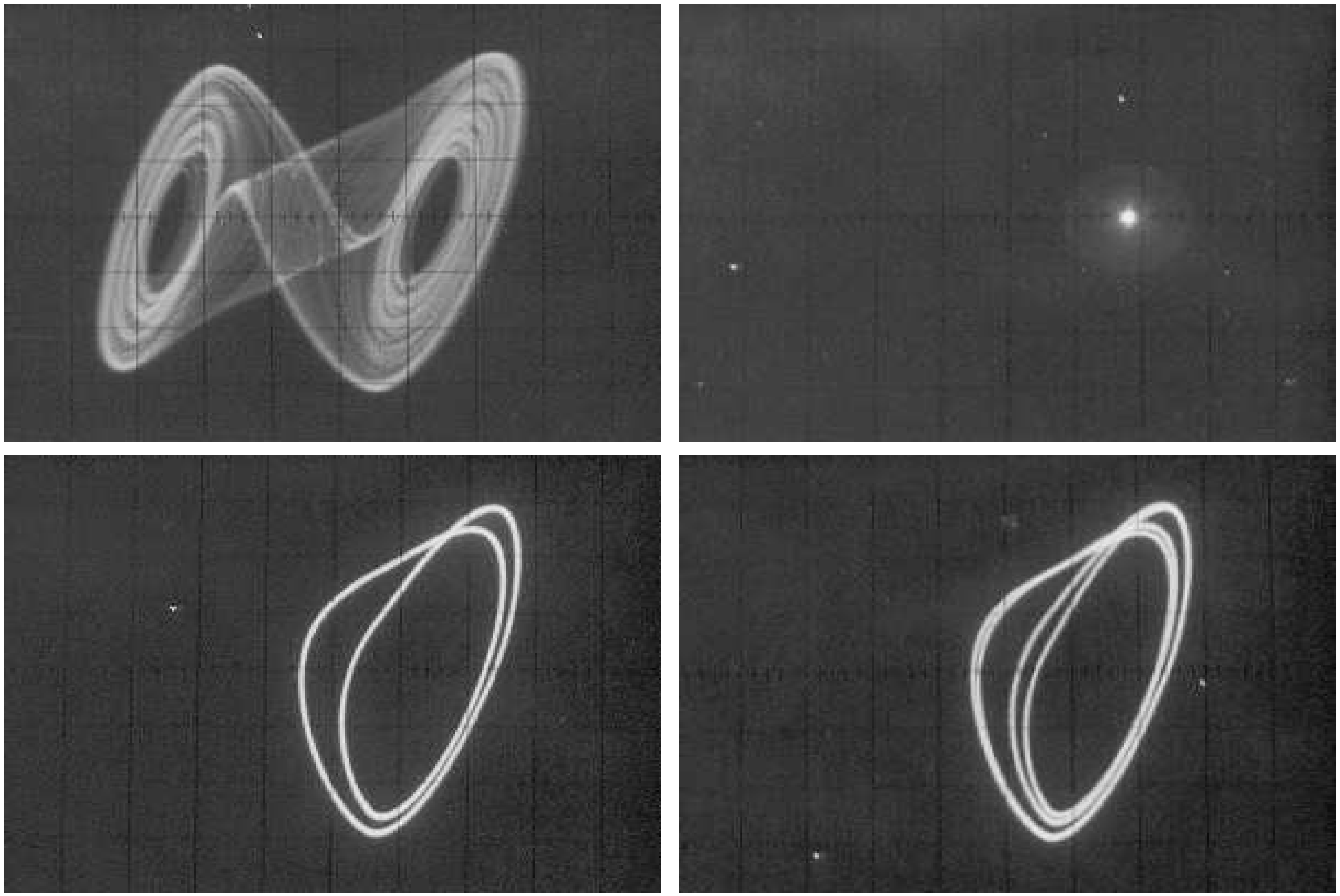
(3) $$\dot{z} = -\beta y$$

The piecewise linear function

$$g(x) = bx + \frac{1}{2}(a - b)(|x + 1| - |x - 1|)$$

Parameters: $\alpha = 10.$, $\beta = 14.87$, $a = -1.27$ and $b = -0.68$

Thresholding Chua's Circuit



Murali and Sinha, Physical Review E, 2003

Hyperchaotic electrical circuit

Constitutes a stringent test of the control method since the system posseses more than one positive lyapunov exponent, and so more than one unstable eigendirection has to be reigned in by thresholding a single variable.

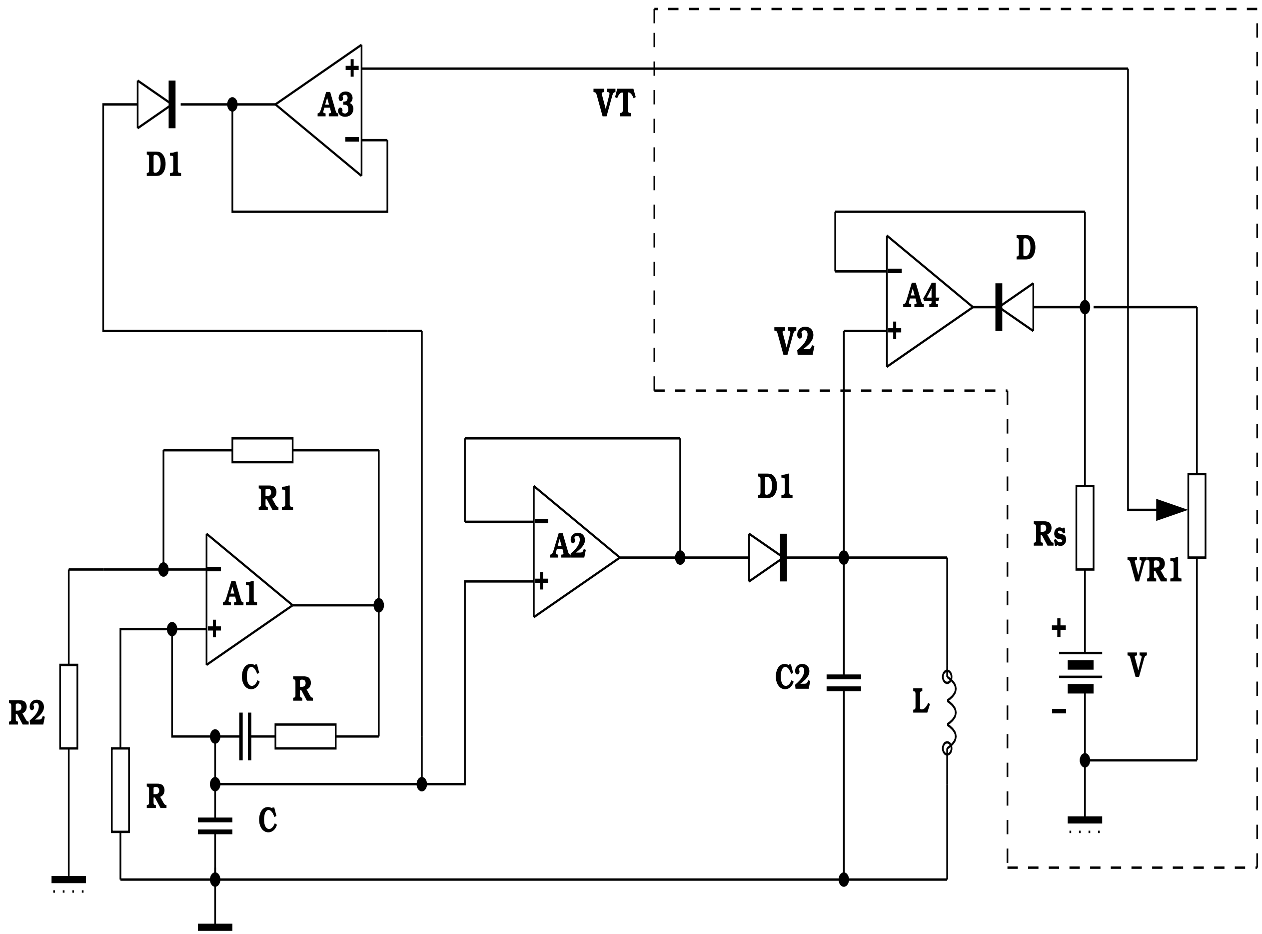
Consider the realisation of four coupled nonlinear (rescaled) ODEs of the form:

$$\begin{aligned}\dot{x_1} &= (k-2)x_1 - x_2 - G(x_1 - x_3)\\ \dot{x_2} &= (k-1)x_1 - x_2\\ \dot{x_3} &= -x_4 + G(x_1 - x_3)\\ \dot{x_4} &= \beta x_3\end{aligned}$$

where

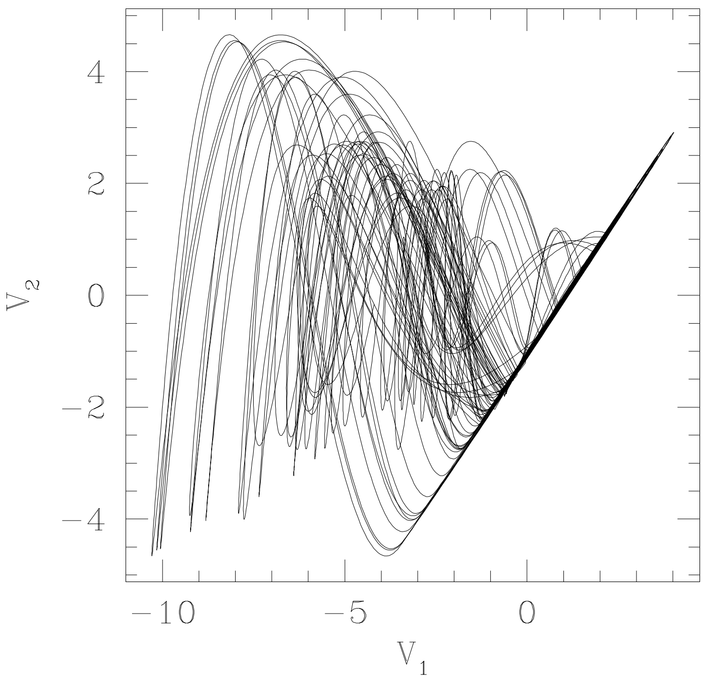
$G(x_1 - x_3) = \frac{1}{2}b[|x_1 - x_3 - 1| + (x_1 - x_3 - 1)]$

with $k = 3.85, b = 88$ and $\beta = 18$

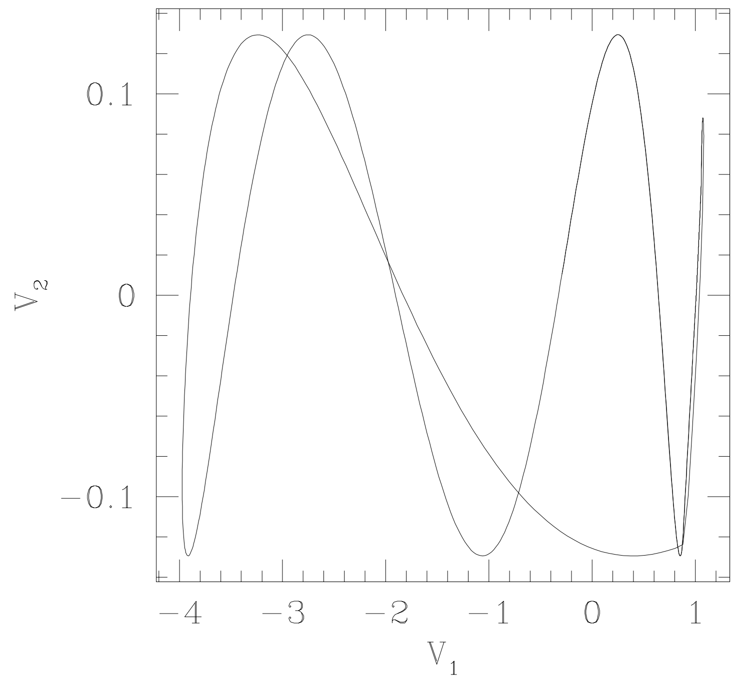


D1
A3
VT
D
A4
V2
R1
A2
D1
A1
Rs
VR1
C
R
C2
L
V
R2
R
C

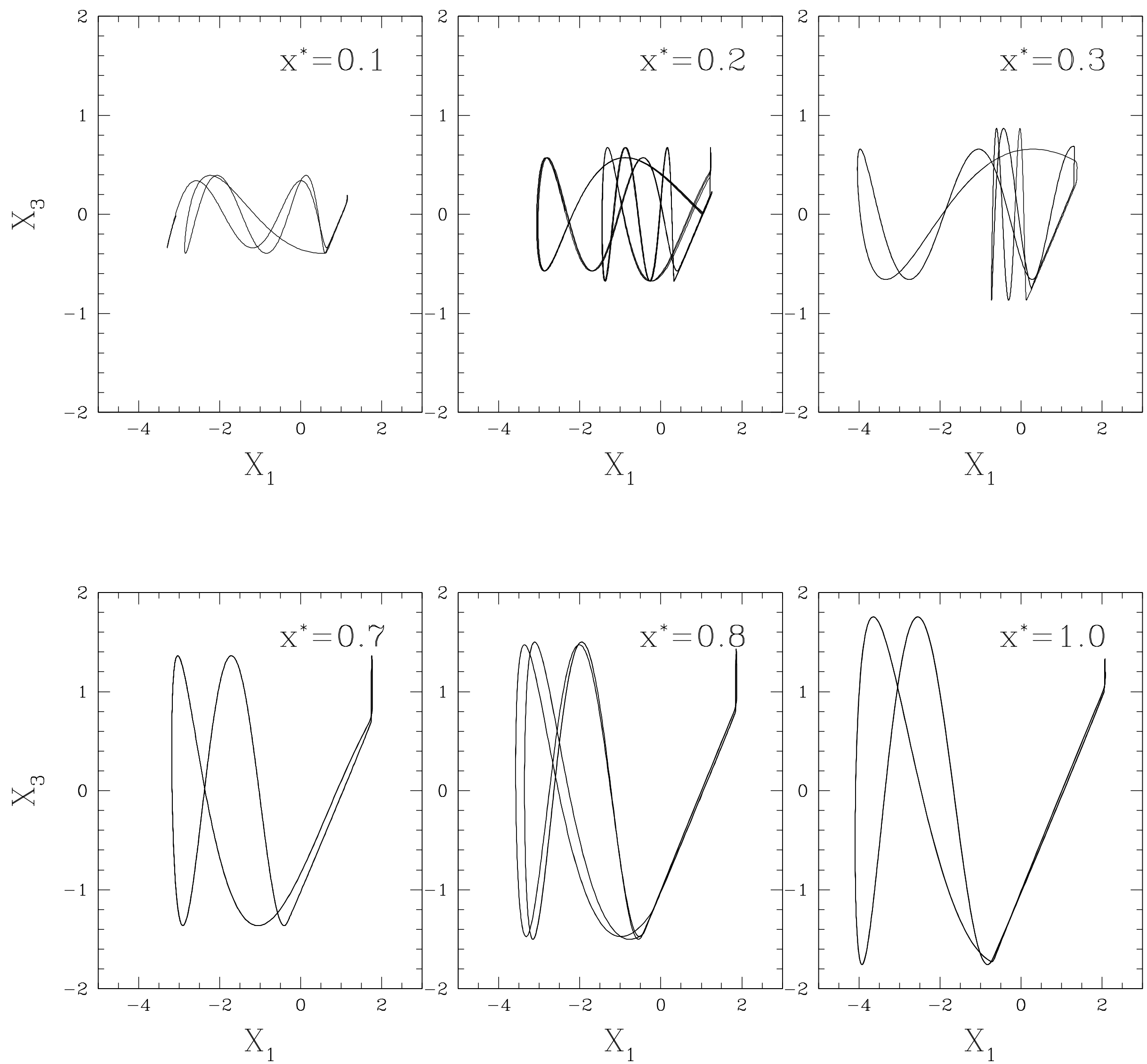
Hyper Chaotic Attractor



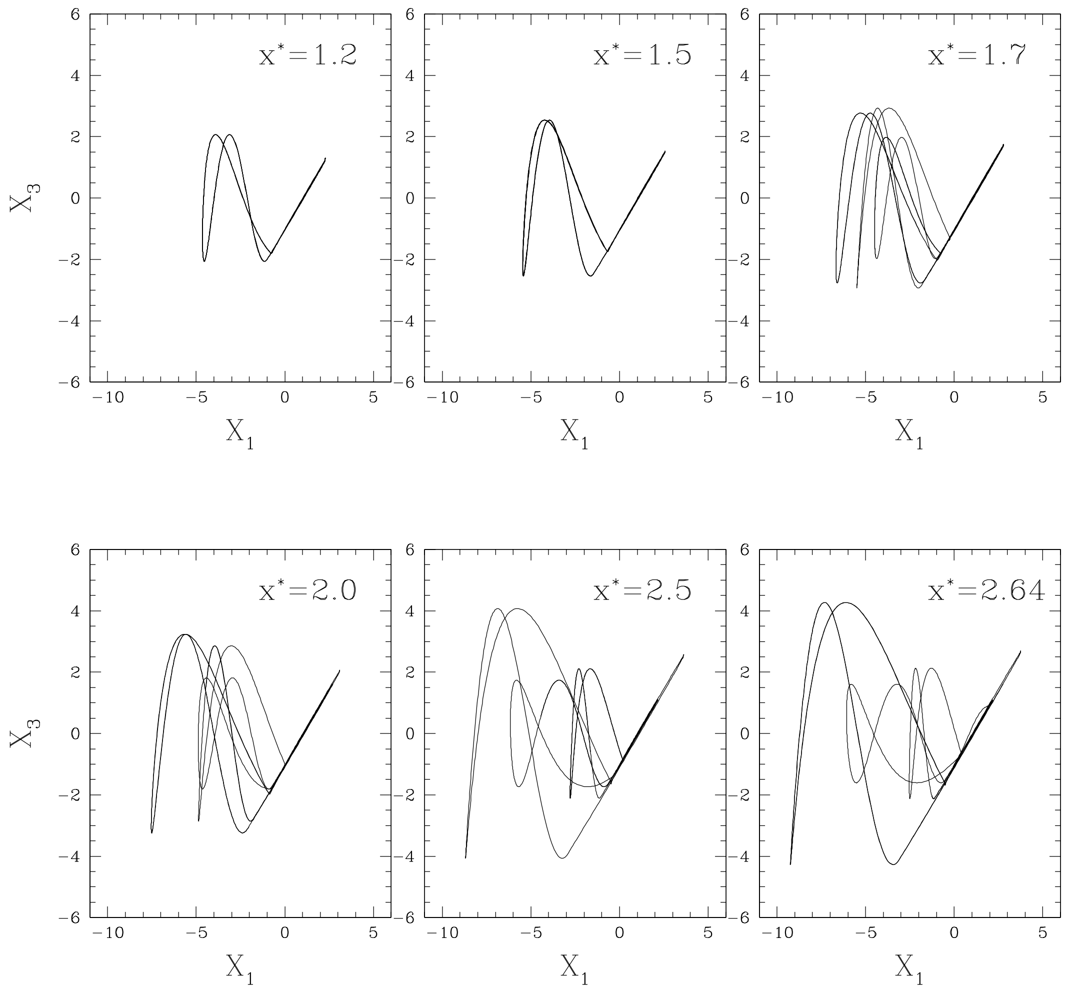
Controlled Orbit



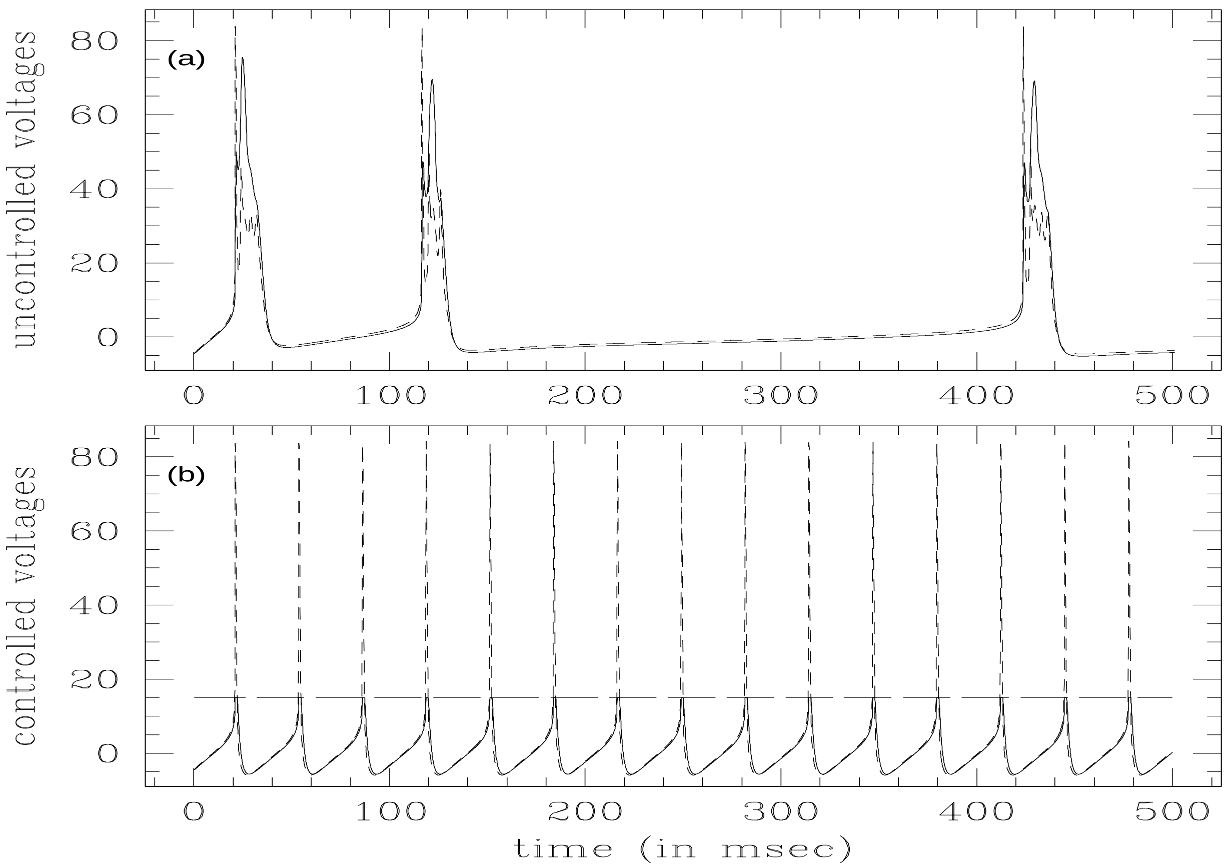
Murali and Sinha, Physical Review E, 2003


x*=0.1
x*=0.2
x*=0.3
x*=0.7
x*=0.8
x*=1.0
X1
X3

Simple Thresholding selects out a very wide variety of patterns even in hyperchaotic systems



Pinsky-Rinzel Neuron : Controlling Spiking
8 coupled ODEs : thresholding one variable



Sinha and Ditto, Physical Review E, 2001

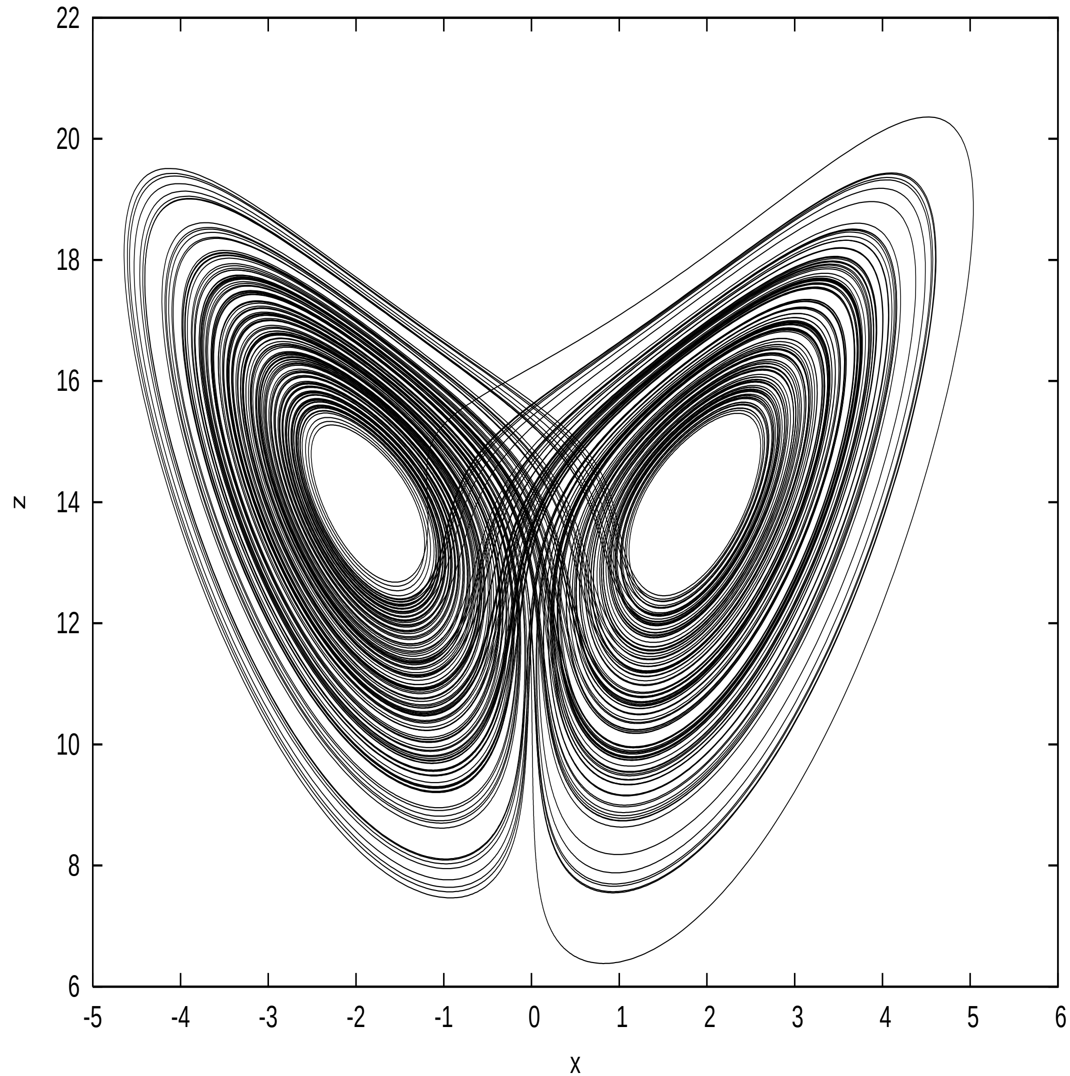
Laser System:

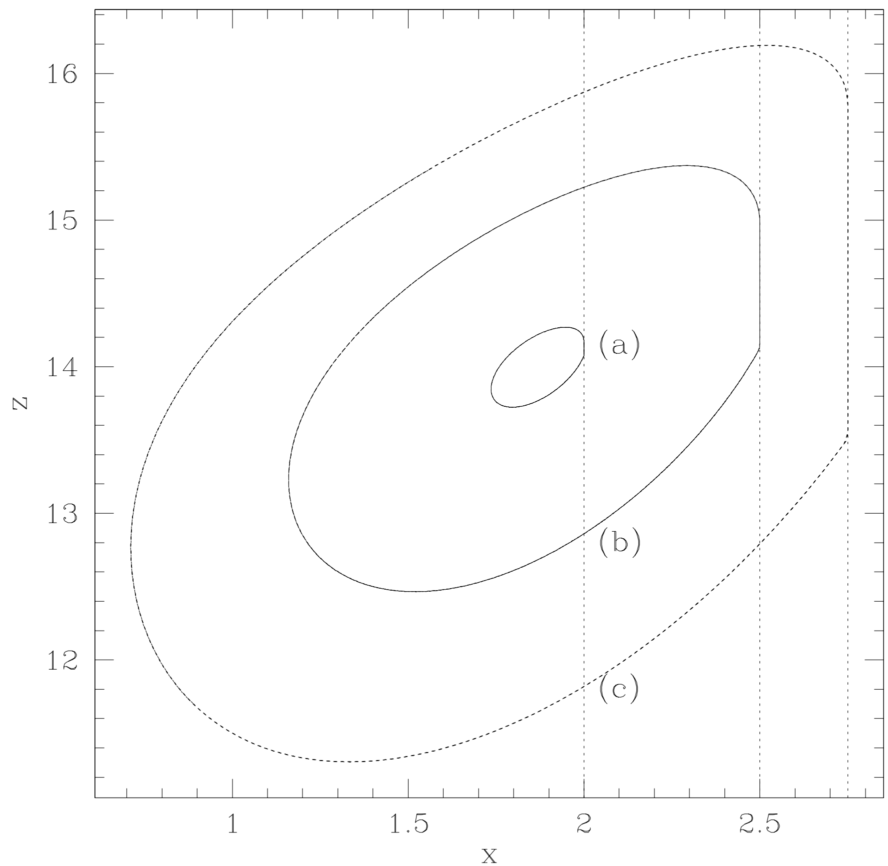
$$
\begin{aligned}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= rx - y - xz \\
\dot{z} &= xy - bzr
\end{aligned}
$$

z variable corresponds to the normalized inversion
x and y variables correspond to normalized amplitudes of the electric field and atomic polarisations

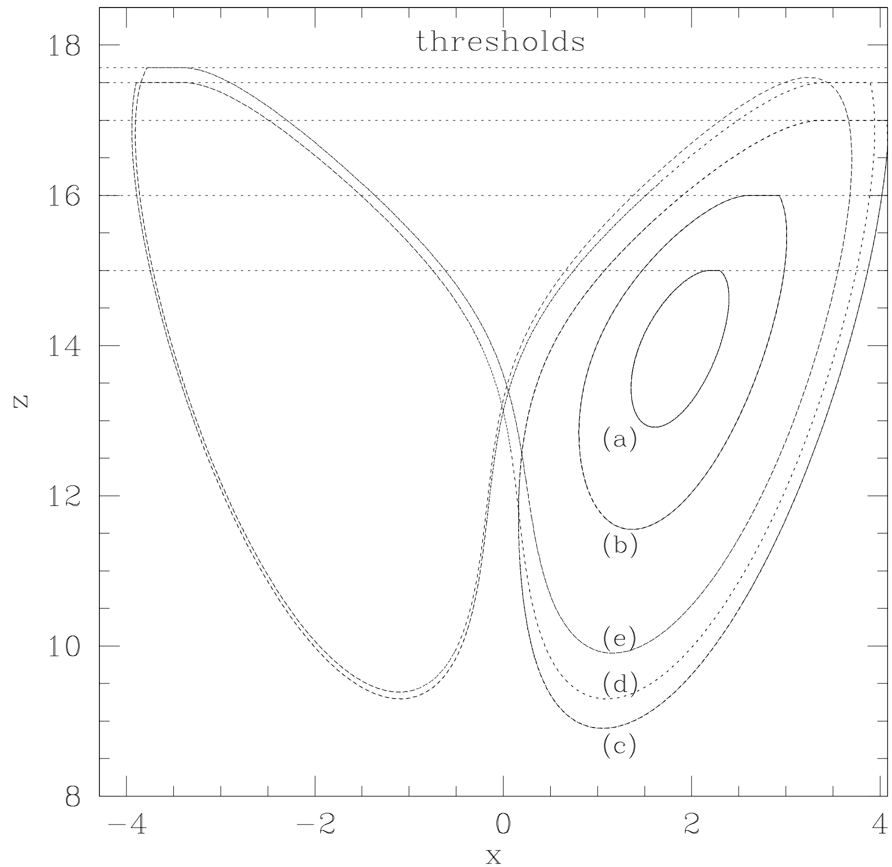
Parameter values, obtained by detailed comparison with experiments, for the corresponding coherently pumped far-infrared ammonia laser system are: $\sigma = 2$, $r = 15$ and $b = 0.25$

Laser System: Lorenz-like Attractor



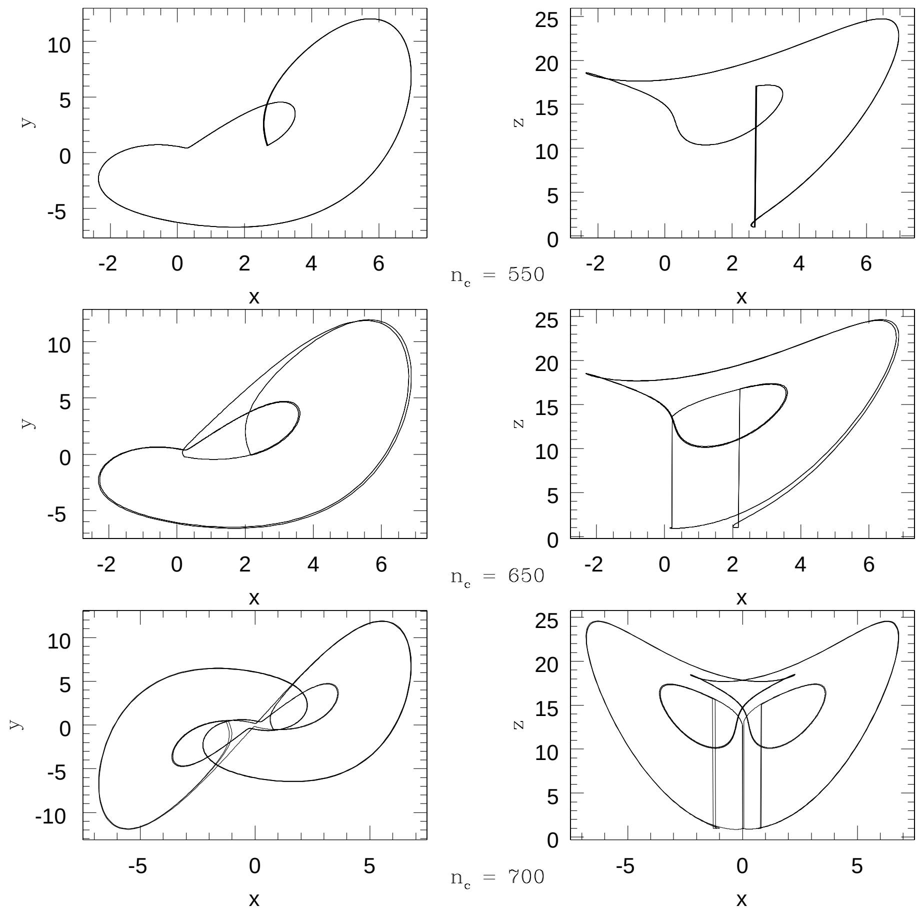


16
15
14
13
12
z
1
1.5
2
2.5
x
(a)
(b)
(c)



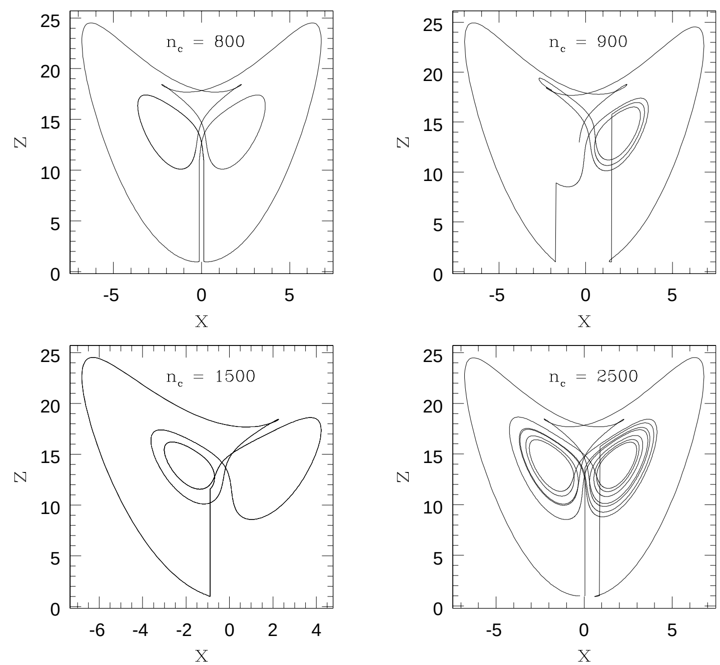
Sinha and Ditto, Physical Review E, 1999

Thresholding at Varying Intervals



Chaotic Ammonia Laser

Varying control intervals offers flexibility in selecting different patterns



Sinha, Physical Review E, 2001

Opportunities offered by Chaos

- Determinism : allows reverse engineering

Opportunities offered by Chaos

- Determinism : allows reverse engineering
- Richness of temporal behaviour : can be used to obtain a wide range of temporal patterns

Opportunities offered by Chaos

- Determinism : allows reverse engineering
- Richness of temporal behaviour : can be used to obtain a wide range of temporal patterns
- Large range of controlled responses : Obtained from very simple mechanisms

Application of thresholding as a strategy for extracting a wide range of temporal patterns from a chaotic system in a controlled manner :

Exploiting Chaos to Design Flexible Hardware

A new direction in harnessing chaos:

- Chaos provides a rich variety of behaviors :

  Can serve as a versatile pattern generator

- Exploit this flexibility for implementing computational tasks

Chaos for Computation

Hardware : Threshold activated chaotic elements
Chaotic Chip, Chaotic Processor

Programming these elements consists of fixing thresholds
such that some desired operation is performed
i.e. certain I/O relations are satisfied

Sinha & Ditto, Physical Review Letters, September 1998
Physcial Review E, 1999

Aim :

Implement all the basic logic gates flexibly using a chaotic element

With the ability to switch between different operational roles

This will allow a more dynamic architecture

Serve as ingredients of a general purpose device more flexible than statically wired hardware

Demonstrate the direct implementation of all the logic gates which are basic and sufficient components of computer architecture today

Sinha, Munakata & Ditto, Phys. Rev. E, 2002

Munakata, Sinha & Ditto, IEEE Trans. on Circuits and Systems, 2002

Flexible implementation : the same chaotic processor can serve as any of the gates by simple change of threshold

Inputs : State of the chaotic element $x \quad \rightarrow \quad x_0 + I_1 + I_2$

Output : Obtained by Threshold Mechanism after Chaotic Update

$$O = f(x) - x^{\star} \qquad \text{if} \quad f(x) > x^{\star}$$
$$O = 0 \qquad \text{if} \quad f(x) < x^{\star}$$

Necessary and Sufficient conditions to be satisfied simultaneously

AND	OR	XOR
$f(x_0) \leq x^*$	$f(x_0) \leq x^*$	$f(x_0) \leq x^*$
$f(x_0 + I) \leq x^*$	$f(x_0 + I) - x^* = I$	$f(x_0 + I) - x^* = I$
$f(x_0 + 2I) - x^* = I$	$f(x_0 + 2I) - x^* = I$	$f(x_0 + 2I) \leq x^*$

NAND	NOT
$f(x_0) - x^* = I$	$f(x_0) - x^* = I$
$f(x_0 + I) - x^* = I$	$f(x_0 + I) \leq x^*$
$f(x_0 + 2I) \leq x^*$	

Robust solutions exist

Operation	AND	OR	XOR	NAND	NOT
x_0	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{2}$
x^*	$\frac{3}{4}$	$\frac{11}{16}$	$\frac{3}{4}$	$\frac{11}{16}$	$\frac{3}{4}$

Richness of the dynamics allows one to select out all the different requisite responses from the same module

Scheme has been experimentally verified

Flexible Dynamic Logic Cell :

Simple mechanism allows one to switch with ease between behaviours emulating different logic gates

This provides sufficient ingredients for directly and flexibly implementing all operations

Universal General Purpose computing device

More versatile than static hardware

Contrast with periodic elements:

It is not possible to extract all the different logic responses from the same element in case of periodic components, as the temporal patterns are inherently limited.

Contrast with random elements:

One cannot design components : need determinism for reverse engineering

Only Chaotic dynamics enjoys both

richness
and
determinism

So one can select out all the different temporal responses necessary to obtain all the different logic patterns with a single evolution function

This ability allows us to construct flexible hardware

Programmable hardware ; Re-configurable hardware

Building blocks of a Dynamical Logic Architecture

- Pre-determined dynamic logic configuration
- Out-come dependent dynamic logic configuration

Possibility of the hardware design evolving during the computation

Threshold control enables us to exploit the richness of chaos in a direct and efficient manner

Used clipped chaos as a pattern generator for the development of a flexible logic module